Adaptation languages for learning: the CAM meta-model

Hendrix, M. , Cristea, A.I. and Stewart, C.

Original citation & hyperlink:
Hendrix, M. , Cristea, A.I. and Stewart, C. (2009). Adaptation languages for learning: the CAM meta-model. In I. Aedo, N-S. Chen Kinshuk, D. Sampson, & L. Zaitseva (Eds). *IEEE International Conference on Advanced Learning Technologies (ICALT 2009)* (pp. 104-106). Institute of Electrical and Electronics Engineers.
http://dx.doi.org/10.1109/ICALT.2009.10

Adaptation languages for learning: the CAM meta-model

Maurice Hendrix[1], Alexandra I. Cristea[1] and Craig Stewart[2]
[1]*The University of Warwick, UK, {maurice, acristea}@dcs.warwick.ac.uk*
[2]*Queen Mary University of London, UK. craig.stewart@elec.qmul.ac.uk*

Abstract

Adaptive Hypermedia (AH) can offer a richer learning experience, tailored to students' needs. However, authoring of AH is complex. Several models and systems have been developed. However the adaptation that can be defined, is limited in terms of (re)-usability by educational authors. To simplify adaptive behaviour authoring, the solution considered most intuitive [7] is a visual environment. In this paper we specify the design of a visual authoring tool. We describe the implications this visual aspect has for the adaptation languages. This reflects lessons learnt from the experience of authoring project partners of the EU GRAPPLE project, hence is of interest to any designer or developer of reusable, general purpose adaptive e-learning system.

1. Introduction

Adaptive Hypermedia (AH) is perceived to have the potential to offer a richer learning experience, personalised for each learner. To realise this potential however, authors need to have the ability to easily create adaptive material. There exist several Adaptive Hypermedia reference models, like AHAM [9] and LAOS [4] Even when using the tools developed based upon these models, authoring remains a difficult and time consuming task for everyone but experts on the specific systems. The problem arises that whilst authors may be domain experts, they are rarely system experts.

In order to simplify adaptive behaviour authoring for the average educational author, the simplest solution [7] is a *visual environment*.

Therefore our approach is graphical in nature; however, we allow for the distinction between content and pedagogical specifications, as well as multiple layers. This has some implications in terms of internal and external description of the language(s) involved, as follows. First of all, a visual language is necessary to serve as interfacing mechanism directly with the author. Moreover we need another language for storage.

We have previously introduced the CAM model (Conceptual Adaptation Model) [6], a generic model for authoring that allows adaptation to be defined in a graphical, flexible and user friendly way.

The remainder of the paper is organized as follows. Section 2 introduces the design of the Authoring tool. In section 3 we describe the required novel adaptation languages,. In section 4 we discuss other related work. Finally, in section 5 we conclude and point to further work.

2. The GRAPPLE Authoring Shell tool

The GRAPPLE Authoring tool is a tool in which educational authors will be able to specify CAM instance models (as such CAM is a meta-modelling system). The author should be able to define all layers of their specific CAM model with the tool, without the need of extensive training in the computer science subjects of (Adaptive) Hypermedia or even programming.[1]

We have decided to split the tool into components corresponding to the model's three mandatory layers [7]; 1) *domain model* (DM) editing (creation of the content, as well as of the concepts representing the content, and the various ways in which the content parts can be linked). 2) *concept relationship* (CRT) editing (editing of pedagogical links, this is in addition to the DM links mentioned above; e.g., prerequisite link, or inhibitor link, etc.) and 3) CAM editing (visual adaptive behaviour editing in a drag &drop environment). A shell is defined to integrate these components into a single tool. This section describes the common infrastructure for this integration. The shell provides three main parts:

1. The *Shell* class that implements the shell;

[1] Please note that some minimal level of hands-on training may still be required, as is commonplace with new technologies in the educational domain.

2. The *AbstractTool* class, an abstract class that tools need to be defined as subclass of, in order to be able to work with the shell;
3. The configuration file, *config.xml,* where configuration settings for the tools are stored, such as the content -, CRT - and CAM repositories.

3. The CAM Languages

As discussed in section 1, we aim to provide educational authors with an easy to use and flexible authoring environment. The authoring solution most appropriate for this goal is a graphical authoring interface. To describe adaptive behaviour using a stepwise approach, authoring languages are needed for the three steps. To this end we define the following languages:

1. The *CAM visual language*, the drag and drop language for authors to create adaptive stories.
2. The *CAM Internal language*, for internal storage of the, possibly incomplete, Adaptive Story line creted by the author.export, storage and reference purposes.
3. The *XML External language*. The language for export of the created Adaptive Course.

This apparent profusion of languages will allow for authors with varying proficiency levels to describe adaptation, as will be explained in the following sections. The rest of this section will discuss the initial design for each of these languages in more detail.

3.1. CAM Visual Language

The CAM visual language is the *drag and drop* language for authors to create adaptive stories. It expresses the CAM model in a visual way and is the only language that is intended for the non-programmer author to work with. This language is essential for ensuring the lowest possible threshold in authoring of adaptation. An example of a CAM description in the visual language is shown in Figure 1. (see also [6]).

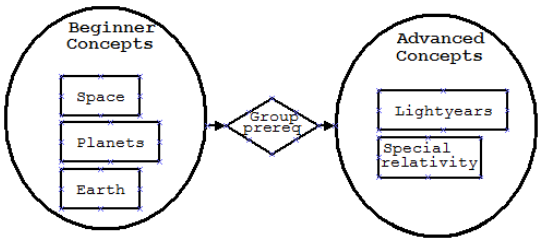

Figure 1 Example of Grouping

Figure 1 shows an example of grouping five concepts in a complex prerequisite structure (where the first three are prerequisites of the last two). The adaptation behaviour complexity is hidden behind the CRT definition. In the CAM tool, everything is performed via drag&drop: thus, CRTs such as the 'Group prereq' in the figure are dragged from a list of CRTs, and then populated with domain concepts, such as 'Space', 'Planets', etc.

3.2. CAM Internal Language

The CAM Internal language is the language used internally by the CAM tool. It is designed to store Adaptive Courses that can be incomplete, and is mainly an XML serialisation of the Graphical Language.

Prior experience with the LAG language [3] shows that non-programmer authors prefer not to be at all involved at the level described by this language. On the other hand, programmers or authors with a Computer Science background prefer the more 'hands-on' experience. Thus, for the latter authors only, the CAM XML language could be potentially used directly to describe adaptive behaviour. Also prior research showed that an XML-based language is preferable, as it is both more portable, and perceived as easier to manipulate than a pure programming language, as in the development of the LAG-XLS language **Error! Reference source not found.**. As the LAG-XLS language was only aimed at learning styles, and we wish to adopt a wider scope of adaptivity in the GRAPPLE project, a new language has to be created.

Below we illustrate the new CAM XML language via an example. We see this type of CAM output as XML descriptions of groups of concepts and *named*, *typed* relations between them. For example, if concept 'Space' is to be seen before concept 'Lightyears', a CAM would be:

```
<CAM>
 <description>Astronomy lesson</description>
 <CRT:prerequisite card = "2">
  <CRT:group name="source" card="3">
    <concept> Space</concept>
  </CRT:group>
  <CRT:group name="target" card="2">
    <concept> Lightyears</concept>
    </CRT:group>
  </CRT:prerequisite>
</CAM>
```

In order to add all the concepts as in Figure 1 we would only need to add concepts 'Planets' and 'Earth' to the first group in the prerequisite relation, and concept 'Special relativity' to the second group. The behaviour semantics of the CRT called 'prerequisite' needs to be separately imported from the CRT repository (where the CRT editing tool is storing these descriptions). Thus, the CRT namespace is used in the

description (these languages are combined similar to *OWL* and *RDF* as per *www.w3.org /*).

3.3. CAM External Language

The CAM External language is the main portable format for the output of the CAM tool. It is designed to be used by other systems (e.g. delivery ststems), allowing them to interface with the CAM tool. The CAM External language consist

- All Domain Models that are used in the course
- All relationships (CRT instances) that are used in the course
- A pruned version of he CAM Internal language, which does not allow parts to be missing.

4. Related work

As the authoring problems facing AH and AEH are both complex and formidable there has been a great deal of research into addressing these issues [3], [8], [9]. The AHA! Graph editing tool and the LAG grammar were both significant advances when they were proposed, but they do not reflect the current state of the art in adaptive technologies. With the GRAPPLE project focused on consolidating AH research and producing global standards, the CAM meta-model represents a timely and fresh examination of adaptive authoring.

Whilst there are no other efforts into the consolidation of AH authoring, in [5] and [2] frameworks for the integration and interoperability of user profiles (learner models) are introduced. However, whilst a user model is a vital part of any AH system, it is only a limited part, the CAM languages are able to process all aspects of such a system. In **Error! Reference source not found.** an AH is described that consists of a variety of web services, this approach allows for ease of extension but is not as versatile as the CAM model.

5. Conclusions and further work

In this paper we have highlighted the need for easier to use and more flexible authoring tools for Adaptive Hypermedia, which will not require extensive training in computer science related subjects. We have seen that the preferred solution is the use of a graphical editor for the author. We have specified the design of the visual authoring tool

Moreover, we have defined the adaptation languages that are required for this kind of approach:

1. The language that serves as interfaces between the *human author and the machine*, as in the case of the CAM visual language, as well as
2. an Internal storage language
3. and the external language to interface with other systems such as delivery systems.

The next steps in this process will involve the development of the authoring tools, as well as the tools for translating CAM instances into usable adaptive courses. From a research perspective, *model definition* and *adaptation language development*, especially on a larger scale, such as the GRAPPLE EU project (involving 15 European partners), brings us closer to the aim of having largely adopted, reusable adaptation specifications, and ultimately, standards.

Acknowledgements

The work presented here has been sponsored by the GRAPPLE EU FP7 project.

6. References

[1] De Bra, P., Smits, D., Stash, N., The Design of AHA! , Proceedings of the ACM Hypertext Conference, Odense, Denmark, pp. 133, 2006.

[2] Celik, I., Stewart, C., Ashman, H.. Interoperability as an Aid to Authoring: Accessing User Models in Multiple AEH Systems, in the 4th Adaptive and Adaptable Educational Hypermedia workshop at the Adaptive Hypermedia (AH) conference, Dublin, Ireland, , Springer, 2006.

[3] Cristea, A.I., Calvi, L., The three Layers of Adaptation Granularity, UM'03, International Conference on User Modelling, Pittsburgh, US, Springer, pp. 145, 2003.

[4] Cristea, A.I., de Mooij, A., LAOS: Layered WWW AHS Authoring Model and their corresponding Algebraic Operators, WWW'03, Alternate Track on Education, Budapest, Hungary. 2003.

[5] Dolog P., Schäfer.M.. A framework for browsing, manipulating and maintaining interoperable learner profiles. UM 2005, volume 3538 of *LNAI*, pp. 397-401, Edinburgh, Scotland, UK, Springer, 2005.

[6] Halasz, F., Schwartz, M. .The Dexter hypertext reference model: Hypermedia. Communications of the ACM, 37(2), pp. 30-39, 1994.

[7] Hendrix, M., De Bra, P., Pechenizkiy, M., Smits, D., Cristea, A.I., Defining adaptation in a generic multi layer model: CAM: The GRAPPLE Conceptual Adaptation Model, In Proceedings of ECTEL, pp. 132-143, 2008.

[8] Ohene-Djan, J. A Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction. PhD thesis, Department of Computing, Goldsmiths College, University of London. 2000.

[9] Wu, H., A Reference Architecture for Adaptive Hypermedia Applications, doctoral thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2, 2001.